sensors

MDPI

*Article*

# An Improved Calibration Method for the IMU Biases Utilizing KF-Based AdaGrad Algorithm

**Zeyang Wen** [†], **Gongliu Yang** [†] **and Qingzhong Cai** [*]

School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China;
cruyffwen@163.com (Z.W.); yanggongliu@buaa.edu.cn (G.Y.)

[*] Correspondence: qingzhong_cai@163.com; Tel.: +86-8233-8112

[†] These authors contributed equally to this work.

**Abstract:** In the field of high accuracy strapdown inertial navigation system (SINS), the inertial measurement unit (IMU) biases can severely affect the navigation accuracy. Traditionally we use Kalman filter (KF) to estimate those biases. However, KF is an unbiased estimation method based on the assumption of Gaussian white noise (GWN) while IMU sensors noise is irregular. Kalman filtering will no longer be accurate when the sensor's noise is irregular. In order to obtain the optimal solution of the IMU biases, this paper proposes a novel method for the calibration of IMU biases utilizing the KF-based AdaGrad algorithm to solve this problem. Three improvements were made as the following: (1) The adaptive subgradient method (AdaGrad) is proposed to overcome the difficulty of setting step size. (2) A KF-based AdaGrad numerical function is derived and (3) a KF-based AdaGrad calibration algorithm is proposed in this paper. Experimental results show that the method proposed in this paper can effectively improve the accuracy of IMU biases in both static tests and car-mounted field tests.

**Keywords:** inertial measurement unit (IMU) calibration; strapdown inertial navigation system (SINS); Kalman filter; gradient descent

check for updates

**Citation:** Wen, Z.; Yang, G.; Cai, Q. An Improved Calibration Method for the IMU Biases Utilizing KF-Based AdaGrad Algorithm. *Sensors* **2021**, *21*, 5055. https://doi.org/10.3390/s21155055

Academic Editor: Felipe Jiménez

Received: 26 June 2021
Accepted: 22 July 2021
Published: 26 July 2021

## 1. Introduction

Strapdown Inertial Navigation Systems (SINS) have been widely used in many aspects of navigation [1]. The SINS has some irreplaceable advantages, such as high autonomy, and it provides continuous and comprehensive navigation information. Therefore, SINS is widely applied in ships and airplanes [2]. For the purpose of realizing the alignment and navigation algorithms of the SINS. The IMU errors have to be calibrated prior to utilization. In an ideal IMU, accelerometers, and gyroscopes coordinate with the same orthogonal sensitivity axes, while the scale factor converts the digital quantity measured by each sensor into the real physical quantity (accelerations and angular rates) [3,4]. However, the IMU is usually affected by non-accurate sensor biases [5], the navigation errors (attitude, velocity and position errors) of the SINS diverge with time [6]. Therefore, the calibration of biases in IMU is a very effective way to enhance the navigation accuracy of SINS.

In [7], Han et al. proposed a method for bias calibration using a single-axis turning table. The disadvantage of this method is that the calibration accuracy depends highly on the turntable accuracy. In current SINS implementations, in order to overcome the non-observability of the biases of the IMU biases, an analytic calibration of IMU biases with two-position is utilized widely [8]. Many works have been done in the field of IMU calibration. In [9], an eight-position self-calibration method was proposed. Emel'yantsev et al. calibrated in-rum drifts of SINS with uniaxial modulation rotation of measurement unit. Since the low end IMU model is nonlinear, some researchers use an adaptive Kalman filter to estimate IMU biases [10–15]. Wang et al. utilized a new online calibration method for integrated navigation systems [16]. In [17], a rotation test was utilized to enhance the observability of KF in order to calibrate the IMU biases. However, KF is an unbiased estimation method based on the assumption of GWN while IMU sensors noise is irregular [18–20]. In [21], an autocovariance least-squares (ALS) technique was proposed to estimate the process noise of IMU. In [22], particle swarm method was utilized in gyro drift estimation. For high-dimensional systems especially for those with weak observability. The adaptive estimation of IMU biases tend to diverge. In order to solve the problem, further researches are needed on this issue.

Allan variance has been widely utilized in the analysis of gyroscopes [23]. Allan variance can provide different types and magnitude information on various noise terms. Due to analogies to IMU sensors, Allan variance has been adapted to the random-drift characterization of a variety of inertial devices [24–26]. Allan variance is a method of representing the root mean square (RMS) random-drift errors as a function of averaging time. However, the equivalent white noise characteristics do not equal the sum of the various noise characteristics analyzed by Allan variance. Thus, the method of using Allan variance to estimate IMU biases is not feasible.

In order to solve the above problems, a KF-based AdaGrad calibration algorithm is proposed. Gradient descent is an iterative optimization algorithm for finding the minimum value of the objective function. For a complex dynamical system, the gradient descent method is utilized to obtain the minimum value of the constructed cost function. In [27–30], researchers proposed a gradient descent approach to solve complex dynamic system problems. The difficulty of the gradient descent method lies in constructing the cost function of a complex dynamic system. In this paper, (1) in order to solve the problem of setting step size, this paper utilized adaptive subgradient methods (AdaGrad); (2) we derive a KF-based AdaGrad numerical cost function in which the objective function is the velocity and position errors; and (3) utilizing the above two theories we propose a novel method called KF-based AdaGrad algorithm. Finally, static tests and field tests are carried out to verify the feasibility and applicability of the investigated method.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of the calibration process of IMU and an analysis of the observability of IMU biases. In Section 3, we derive a KF-based AdaGrad numerical cost function and propose a KF-based AdaGrad calibration algorithm. Rotation tests and car-mounted field tests are carried out in Section 4 to compare the proposed method and the existing calibration method. The conclusions are given in Section 5.

## 2. Two-Position Calibration Modeling

The navigation performance of the SINS relies on a high-precision IMU. Error models have been developed for the IMU in [7,8]. Under the condition of the static base, the velocity of the navigation solution is the velocity error. According to the SINS error transformation model, the misalignment angle error can be derived from the velocity error. Since the inertial navigation system has no obvious movement of the geographical position, the velocity of the inertial navigation system approaches zero. Therefore, in the calibration process, the SINS navigation algorithm can be replaced by some simplified equations.

In a SINS, the IMU is directly mounted on the vehicle or the carrier without a stable platform. The angular velocity and acceleration in the body frame (denoted by $b$) are measured by the gyroscopes and accelerometers. Choose the local geographical frame (East-North-Vertical frame) as the navigation frame (denote by $n$).

Let $\boldsymbol{\omega}_{en} = 0$ to get the simplified attitude algorithm:

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n \left[ (\boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{ie}^b) \times \right] \tag{1}$$

$\boldsymbol{C}_b^n$ is the direction cosine matrix to transform vector from $b$-frame (IMU body orthogonal frame which aligned with Right-North-Up axes) to $n$-frame (navigation frame which aligned with East-North-Vertical axes). The subscript $i$ denotes the inertial-frame.

$\boldsymbol{\omega}_{ib}^{b}$ represents the angular velocity measured by the gyroscope in the $b$-frame. $\boldsymbol{\omega}_{ie}^{b}$ is the self-rotation angular velocity.

Specific force equation as one of the basic equations of the SINS can be written as :

$$\dot{\boldsymbol{v}}^{n} = \boldsymbol{C}_{b}^{n}\boldsymbol{f}_{sf}^{b} - (2\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n}) \times \boldsymbol{v}^{n} + \boldsymbol{g}^{n} \tag{2}$$

where $\boldsymbol{v}^{n} = \begin{bmatrix} v_{E}^{n} & v_{N}^{n} & v_{U}^{n} \end{bmatrix}^{T}$, $g$ is the gravity acceleration of the earth. Let $\boldsymbol{v}^{n} = \boldsymbol{0}$ to obtain the simplified velocity algorithm:

$$\dot{\boldsymbol{v}}^{n} = \boldsymbol{C}_{b}^{n}\boldsymbol{f}_{sf}^{b} + \boldsymbol{g}^{n} \tag{3}$$

Following the above derivation, the attitude error equation and the velocity error equation can be written as:

$$\begin{aligned} \dot{\boldsymbol{\phi}} &= \boldsymbol{\phi} \times \boldsymbol{\omega}_{ie}^{n} - \boldsymbol{\varepsilon}^{n} \\ \delta\dot{\boldsymbol{v}}^{n} &= \boldsymbol{f}_{sf}^{n} \times \boldsymbol{\phi} + \boldsymbol{\nabla}^{n} \end{aligned} \tag{4}$$

where $\boldsymbol{\phi}$ is the misalignment angle vector and $\delta v^{n}$ is the velocity error vector. $\boldsymbol{\varepsilon}^{n}$ is the gyroscope equivalent constant bias and the $\boldsymbol{\nabla}^{n}$ is the accelerometer equivalent constant bias. The expressions for $\boldsymbol{\varepsilon}^{n}$ and $\boldsymbol{\nabla}^{n}$ are written as follows:

$$\boldsymbol{\varepsilon}^{n} = \begin{bmatrix} \varepsilon_{E} \\ \varepsilon_{N} \\ \varepsilon_{U} \end{bmatrix} = \begin{bmatrix} C_{11}\varepsilon_{x}^{b} + C_{12}\varepsilon_{y}^{b} + C_{13}\varepsilon_{z}^{b} \\ C_{21}\varepsilon_{x}^{b} + C_{22}\varepsilon_{y}^{b} + C_{23}\varepsilon_{z}^{b} \\ C_{31}\varepsilon_{x}^{b} + C_{32}\varepsilon_{y}^{b} + C_{33}\varepsilon_{z}^{b} \end{bmatrix} \tag{5}$$

$$\boldsymbol{\nabla}^{n} = \begin{bmatrix} \nabla_{E} \\ \nabla_{N} \\ \nabla_{U} \end{bmatrix} = \begin{bmatrix} C_{11}\nabla_{x}^{b} + C_{12}\nabla_{y}^{b} + C_{13}\nabla_{z}^{b} \\ C_{21}\nabla_{x}^{b} + C_{22}\nabla_{y}^{b} + C_{23}\nabla_{z}^{b} \\ C_{31}\nabla_{x}^{b} + C_{32}\nabla_{y}^{b} + C_{33}\nabla_{z}^{b} \end{bmatrix} \tag{6}$$

where $C_{ij}$ denotes the elements of $\boldsymbol{C}_{b}^{n}$. $\boldsymbol{\varepsilon}^{b}$ and $\boldsymbol{\nabla}^{b}$ are the gyroscope and accelerometer equivalent constant bias.

In the above equation, there is no cross-linking relationship between the Equation ($\delta\dot{\boldsymbol{v}}_{\boldsymbol{U}} = \nabla_{U}$) and the other equations. It can be concluded that the vertical velocity error does not have any effect on the misalignment angle estimation. Therefore, in the analysis of the misalignment angle estimation, the influence of the vertical velocity and vertical accelerometer bias can generally be ignored. By expanding the $\boldsymbol{\varepsilon}^{b}$ and $\boldsymbol{\nabla}^{b}$ to the state, the calibration state space model can be established as follows:

$$\begin{cases} \dot{\boldsymbol{X}} = \boldsymbol{F}\boldsymbol{X} + \boldsymbol{G}\boldsymbol{W}^{b} \\ \boldsymbol{Z} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{V} \end{cases} \tag{7}$$

where

$$\boldsymbol{F} = \begin{bmatrix} -(\boldsymbol{\omega}_{ie}^{n}\times) & \boldsymbol{0}_{3\times3} & -\boldsymbol{C}_{b}^{n} & \boldsymbol{0}_{3\times3} \\ -(\boldsymbol{g}^{n}\times) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{C}_{b}^{n} \\ \multicolumn{4}{c}{\boldsymbol{0}_{6\times12}} \end{bmatrix} \tag{8}$$

$$\boldsymbol{G} = \begin{bmatrix} -\boldsymbol{C}_{b}^{n} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{C}_{b}^{n} \\ \multicolumn{2}{c}{\boldsymbol{0}_{6\times6}} \end{bmatrix} \tag{9}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times6} \end{bmatrix} \tag{10}$$

According to the designed two-position calibration scheme, the actual system is time-varying system. PWCS method and SVD method are both employed for the two-position calibration scheme. The rank of this observability matrix is 7 [26,27]. Five states $\nabla_{E}$, $\nabla_{N}$, $\varepsilon_{E}$, $\varepsilon_{N}$ and $\varepsilon_{U}$ are considered unobservable. In order to improve the observability, two-position calibration is needed. This is equivalent to changing the $\boldsymbol{C}_{b}^{n}$ of the SINS. Furthermore, changing the SINS error equation from a stationary system to a time-varying system, which is beneficial to improve the observability of the IMU biases. The scale factor errors of the three gyros and accelerometers are not observable during the process of two-position calibration. Therefore, we do not consider the scale factor errors in Equation (7).

Using the piece-wise constant system (PWCS) observability analysis method [27], the rank of the observability matrix is 11 except $\varepsilon_U$.

The value of the elements of the process noise matrix [29–31], which is the directly determined by the value of the elements in the $\boldsymbol{W}$ matrix. As is shown in Equation (12), the covariance of the $\boldsymbol{W}$ matrix is $\boldsymbol{Q}$. Furthermore, the accuracy of the KF is determined by the reliability of the $\boldsymbol{Q}$ information. $\boldsymbol{W}$ and $\boldsymbol{Q}$ can be written as:

$$\boldsymbol{W}^b = \begin{bmatrix} w_{gx}^b & w_{gy}^b & w_{gz}^b & w_{ax}^b & w_{ay}^b & w_{az}^b \end{bmatrix}^{\mathrm{T}} \tag{11}$$

$$\begin{cases} \mathrm{E}[\boldsymbol{W}_k] = 0, & \mathrm{E}[\boldsymbol{W}_k \boldsymbol{W}_j^{\mathrm{T}}] = \boldsymbol{Q}_k \delta_{kj} \\ \mathrm{E}[\boldsymbol{V}_k] = 0, & \mathrm{E}[\boldsymbol{V}_k \boldsymbol{V}_j^{\mathrm{T}}] = \boldsymbol{R}_k \delta_{kj} \\ \mathrm{E}[\boldsymbol{W}_k \boldsymbol{V}_j^{\mathrm{T}}] = 0 \end{cases} \tag{12}$$

However, KF is an unbiased estimation method based on the assumption of GWN while IMU sensor noise is irregular. In order to obtain the optimal solution of IMU biases, this paper presents a novel method for the calibration of IMU biases utilizing the KF-based AdaGrad algorithm in Section 4.

## 3. KF-Based AdaGrad Algorithm

### 3.1. Gradient Descent Algorithm

Gradient descent is an iterative algorithm to minimize an objective function. If the multi-variable function $\boldsymbol{F}(\boldsymbol{x})$ is defined and differentiable. Given an initial point, and then follow the negative of the gradient of the function at every iteration. When the point moves to a critical point where the point does not change anymore (or change a value that approaches zero), it has reached the desired local minimum. The basic algorithm is written as follow:

$$\boldsymbol{x}_{\mathbf{n+1}} = \boldsymbol{x}_{\mathbf{n}} - \gamma_{\mathbf{n}} \nabla \boldsymbol{F}(\boldsymbol{x}_{\mathbf{n}}) \tag{13}$$

where $\boldsymbol{x}$ is the variable vector and $\gamma$ is the step size.

However, setting the step size of gradient descent is a key issue. An inappropriate step size may cause the function $\boldsymbol{F}(\boldsymbol{x})$ easily trapped in saddle points. In this paper, the sampling rate of the IMU is 200 Hz. Therefore the sampling time is 0.005 ms. In order to ensure that there are no large values at the beginning. The step size we set is 0.005 at the beginning.

In order to solve this problem, we utilized two methods: line search and adaptive subgradient methods (AdaGrad).

### 3.2. Line Search

Line search is an approach is to adapt the step size at every iteration. The line search will calculate the descent direction $p_k$ and the step size $\alpha_k$ to move in this direction. The following condition must be satisfied: ${p_k}^T \nabla \boldsymbol{F}_k < 0$ so that the function $\boldsymbol{F}(\boldsymbol{x})$ can be guaranteed to fall along this direction. Furthermore, the search direction can be written as: $p_k = -B_k^{-1} \nabla \boldsymbol{F}_k$. The step size $\alpha_k$ should minimize the following function:

$$\phi(\alpha) = \boldsymbol{F}(\boldsymbol{x}_k + \alpha p_k) \tag{14}$$

However, it is difficult to obtain the $\alpha$ that minimizes the above equation, and the amount of calculation is relatively large. The commonly used method is to obtain a larger step size as much as possible with an acceptable amount of calculation, in order to reduce the value of $\phi(\alpha)$ as much as possible. The general line search method consists of the following two steps:

1. Bracketing: Finding a range containing ideal steps.

2. Dichotomy or Interpolation: Using the dichotomy or interpolation to find the step size in this interval.

The KF-based AdaGrad algorithm is very computationally intensive. Even though the above method can reduce a lot of calculations, a faster step-size adaptive approach is needed. Thus, we utilized AdaGrad to solve this problem.

### *3.3. Adaptive Subgradient Methods (AdaGrad)*

The basic idea of AdaGrad is to use different step sizes at every iteration. The step size is large at the beginning for rapid gradient descent. As the optimization process progresses, the step size is reduced for the gradient that has fallen significantly, and the larger step size is maintained for the gradient that has not decreased so much.

For the basic Gradient descent, the vectorized algorithm can be written as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta \nabla \boldsymbol{F}(\boldsymbol{x}_k) \tag{15}$$

AdaGrad multiplies the step size by a parameter that varies with the number of iterations:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \tfrac{\eta}{\sqrt{G_k + \varepsilon}} \nabla \boldsymbol{F}(\boldsymbol{x}_k) \\ \boldsymbol{G}_k &= \boldsymbol{G}_{k-1} + \nabla \boldsymbol{F}(\boldsymbol{x}_k) \end{aligned} \tag{16}$$

where $\eta$ is a small value to prevent the denominator to be zero.

It is easy to discover that as the algorithm continues to iterate, $\boldsymbol{G}_k$ will grow larger, and the step size will grow smaller. The AdaGrad algorithm starts with the convergence of high speed and ends with a small step size.

Since we proposed a KF-based AdaGrad numerical cost function $\boldsymbol{F}(\boldsymbol{x})$, the AdaGrad algorithm needs to be discretized. The Algorithm flowchart is shown in Figure 1.

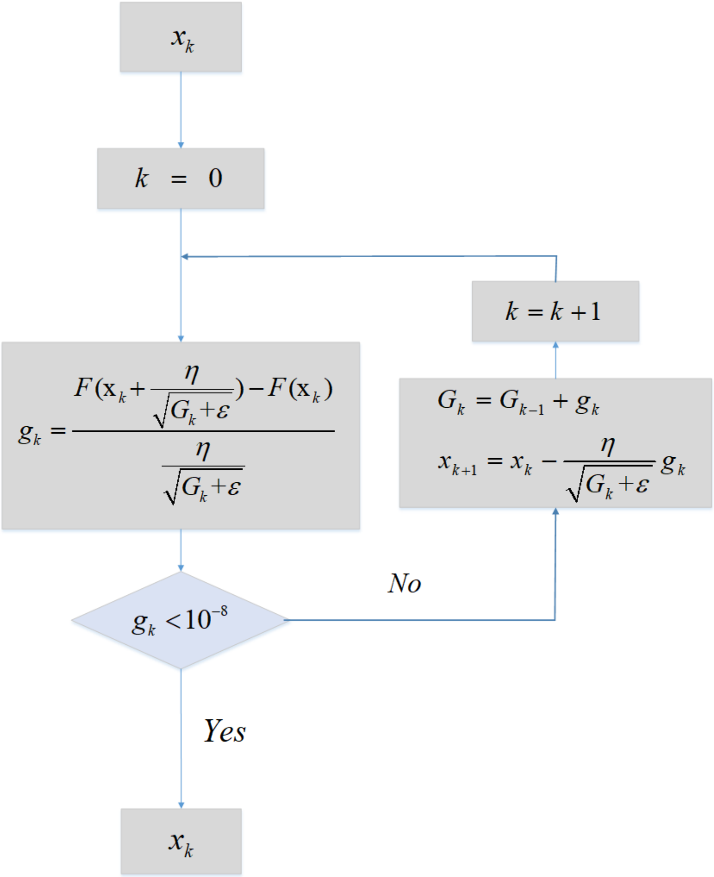
$x_k$
$k = 0$
$k = k+1$
$g_k = \frac{F(x_k + \frac{\eta}{\sqrt{G_k+\varepsilon}}) - F(x_k)}{\frac{\eta}{\sqrt{G_k+\varepsilon}}}$
$G_k = G_{k-1} + g_k$
$x_{k+1} = x_k - \frac{\eta}{\sqrt{G_k+\varepsilon}} g_k$
No
$g_k < 10^{-8}$
Yes
$x_k$

**Figure 1.** Flowchart of AdaGrad.

Given $\boldsymbol{x}_k$ as an initial value and then enters the loop. When the gradient value $\boldsymbol{g}_k$ is close to zero (this paper selects 10e-8 as the critical value). The function jumps out of the loop and the output is $\boldsymbol{x}_k$. In this paper, $\boldsymbol{x}_k = \begin{bmatrix} \varepsilon_x & \varepsilon_y & \varepsilon_z & \nabla_x & \nabla_y & \nabla_z \end{bmatrix}^T$, where $\boldsymbol{Q}_k$ is the process Noise Covariance Matrix. $\boldsymbol{F}(\boldsymbol{x}_k)$ is the KF-based numerical function, it is the function which consists of two database and multiple iterations of one KF, we will derive it in the next subsection.

The psudocode is listed in the Algorithm 1 and is shown as follows:

```
Algorithm 1: AdaGrad algorithm.
    Input: The state vector before AdaGrad algorithm.
    Output: The estimated state vector. Initialization:
    $\boldsymbol{x}_0 = [\ \varepsilon_x \ \ \varepsilon_y \ \ \varepsilon_z \ \ \nabla_x \ \ \nabla_y \ \ \nabla_z\ ]^T, k = 0, G_0 = 0, g_k = 0$
    repeat:
    Calculate $g_k$:
    $\sqrt{G_k + \varepsilon}\boldsymbol{\eta}^{-1}[\boldsymbol{F}\left(\boldsymbol{x}_k + \frac{\boldsymbol{\eta}}{\sqrt{G_k+\varepsilon}}\right) - \boldsymbol{F}(\boldsymbol{x}_k)]$
    Update $G_k$:
    $G_k = G_{k-1} + g_k$
    Update $\boldsymbol{x}_k$
    $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - \frac{\boldsymbol{\eta}}{\sqrt{G_k+\varepsilon}} g_k$
    until $g_k < 1e - 8$
```

### 3.4. KF-Based AdaGrad Calibration Algorithm

AdaGrad is a good method to obtain the minimum value of a function. Setting the IMU biases accurately could improve the accuracy of SINS. In order to obtain the optimal solution of IMU biases, we derives a numerical data function $\boldsymbol{F}(\boldsymbol{x}_k)$. As $\boldsymbol{F}(\boldsymbol{x}_k)$ reaches its minimum value, the IMU reaches its biases optimal solution.

In practical utilization, the effective compensation method of IMU biases is conducted as the following steps:

1. SINS (contains IMU and navigation micro-computer) stiffly fixed on a static platform, power up and start navigation.
2. Since the real velocity is 0 under the static platform condition, the pure inertial navigation output velocity is the velocity error. According to the principle of inertial navigation, the velocity error curve will show the form of Schuler oscillation.
3. In the oscillation process of the velocity error curve, the velocity error and positioning error shows the effect of calibration.

As is shown in Algorithm 1, the estimation of $\boldsymbol{x}_k$ can be written as:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - \frac{\eta}{\sqrt{G_k + \varepsilon}} g_k$$

In summary, if the minimum RMSE of the position error can be obtained, and then the bias compensation effect reaches its optimal. Utilizing a two-position calibration algorithm, IMU biases can be obtained when the process noise matrix is determined. We have derived the KF-based AdaGrad numerical function $\boldsymbol{F}(\boldsymbol{x}_k)$ and show the diagram of the numerical function in Figure 2. The pseudocode of the KF-based numerical function is listed in Algorithm 2 and is shown as follows.

As is shown in Figure 2, the entire KF-based numerical function $\boldsymbol{F}(\boldsymbol{x}_k)$ can be divided into two major processes: (1) two-position calibration process and (2) SINS navigation process. A detailed explanation of these two processes is as follows:

1. A two-position calibration process: based on the calibration database (contains $n1$ sets of data), $\boldsymbol{\omega}^b_{ib(m)}$ and $\boldsymbol{f}^b_{ib(m)}$ can be used for SINS algorithm. According to the theory of Section 3, set IMU biases and using KF to estimate attitude, velocity and position. During every iteration, intermediate variables $\varepsilon_{k(m)}$ and $\nabla_{k(m)}$ are used as feedback

to correct the data retrieved from the database (calibration database) during the next round of iterations. When $m = n1$, the data in the database (calibration database) has been calculated. Then this part can obtain outputs $\varepsilon_k$ and $\nabla_k$.

2. SINS navigation process: based on the SINS static data base (contains n2 sets of data), $\boldsymbol{\omega}^b_{ib(j)}$ and $\boldsymbol{f}^b_{ib(j)}$ can be used for SINS algorithm. Utilizing Equation (17) to calculate the square of position error:

$$RMSE_k = \sqrt{\delta Pos_{k(n2)}}/n_2 \tag{17}$$

When $j = n2$, the process of inertial navigation comes to an end.

After the above process, the numerical function output $RMSE_k$ can be obtained by the input $\boldsymbol{x}_k$. Correspondingly we have derived the KF-based numerical function $\boldsymbol{F}(\boldsymbol{x})$. Bring $\boldsymbol{F}(\boldsymbol{x})$ into the algorithm flow chart shown in Figure 1. Let $\boldsymbol{x}_k = \begin{bmatrix} \varepsilon_x & \varepsilon_y & \varepsilon_z & \nabla_x & \nabla_y & \nabla_z \end{bmatrix}^T$, where $\boldsymbol{x}_k$ is the IMU biases vector value. $\boldsymbol{F}(\boldsymbol{x}_k)$. Afterwards, the minimum $RMSE_k$ can be obtained by the AdaGrad algorithm, and the $\boldsymbol{x}_k$ is the optimal value.

**Algorithm 2:** KF-based numerical function.

```
**Input:** state vector $\boldsymbol{x}_k = [\ \varepsilon_x\ \ \varepsilon_y\ \ \varepsilon_z\ \ \nabla_x\ \ \nabla_y\ \ \nabla_z\ ]^T$
**Output:** position root mean square error $RMSE_k$
**loop 1 (Two position calibration process):**
SINS algorithm:
$\boldsymbol{C}^n_{bk} = \boldsymbol{C}^n_{bk-1}(\boldsymbol{I} + T_s\boldsymbol{\omega}^b_{nb}\times)$
$\boldsymbol{a}^n_{k,k-1} = \boldsymbol{C}^n_{bk-1}\boldsymbol{f}^b_{sfk} - (2\boldsymbol{\omega}^n_{iek-1} + \boldsymbol{\omega}^n_{enk-1}) \times \boldsymbol{v}^n_{k-1} + \boldsymbol{g}^n$
$\boldsymbol{v}^n_k = \boldsymbol{v}^n_k + T_s\boldsymbol{a}^n_{k,k-1}$
$\boldsymbol{Z}_k = \boldsymbol{v}^n_k - \boldsymbol{0}$
Kalman filtering:
$\hat{\boldsymbol{X}}_{k/k-1} = \boldsymbol{\Phi}_{k/k-1}\hat{\boldsymbol{X}}_{k-1}$
$\boldsymbol{P}_{k/k-1} = \boldsymbol{\Phi}_{k/k-1}\boldsymbol{P}_{k-1}\boldsymbol{\Phi}^{\mathrm{T}}_{k/k-1} + \boldsymbol{\Gamma}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{\Gamma}^{\mathrm{T}}_{k-1}$
$\boldsymbol{K}_k = \boldsymbol{P}_{k/k-1}\boldsymbol{H}^{\mathrm{T}}_k(\boldsymbol{H}_k\boldsymbol{P}_{k/k-1}\boldsymbol{H}^{\mathrm{T}}_k + \boldsymbol{R}_k)^{-1}$
$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k/k-1}$
**until** calibration data base is used up.
**return** $\boldsymbol{x}_k$ **end** loop 1
input $\boldsymbol{x}_k$ into loop 2.
**loop 2 (SINS navigation process):**
SINS navigation process:
$\boldsymbol{C}^n_{bk} = \boldsymbol{C}^n_{bk-1}(\boldsymbol{I} + T_s\boldsymbol{\omega}^b_{nb}\times)$
$\boldsymbol{a}^n_{k,k-1} = \boldsymbol{C}^n_{bk-1}\boldsymbol{f}^b_{sfk} - (2\boldsymbol{\omega}^n_{iek-1} + \boldsymbol{\omega}^n_{enk-1}) \times \boldsymbol{v}^n_{k-1} + \boldsymbol{g}^n$
$\boldsymbol{v}^n_k = \boldsymbol{v}^n_k + T_s\boldsymbol{a}^n_{k,k-1}$
$L_k = L_{k-1} + \frac{T_s\tilde{v}^n_{Nk-1}}{R_M+h_{k-1}}$
$\lambda_k = \lambda_{k-1} + \frac{T_s\tilde{v}^n_{Ek-1}\mathrm{secL}_{k-1}}{R_N+h_{k-1}}$
$h_k = h_{k-1} + T_s\tilde{v}^n_{Uk-1}$
calculate $\delta pos_k$
**until** SINS static data base is used up
**return** $RMSE_k$
**end** loop 2
```

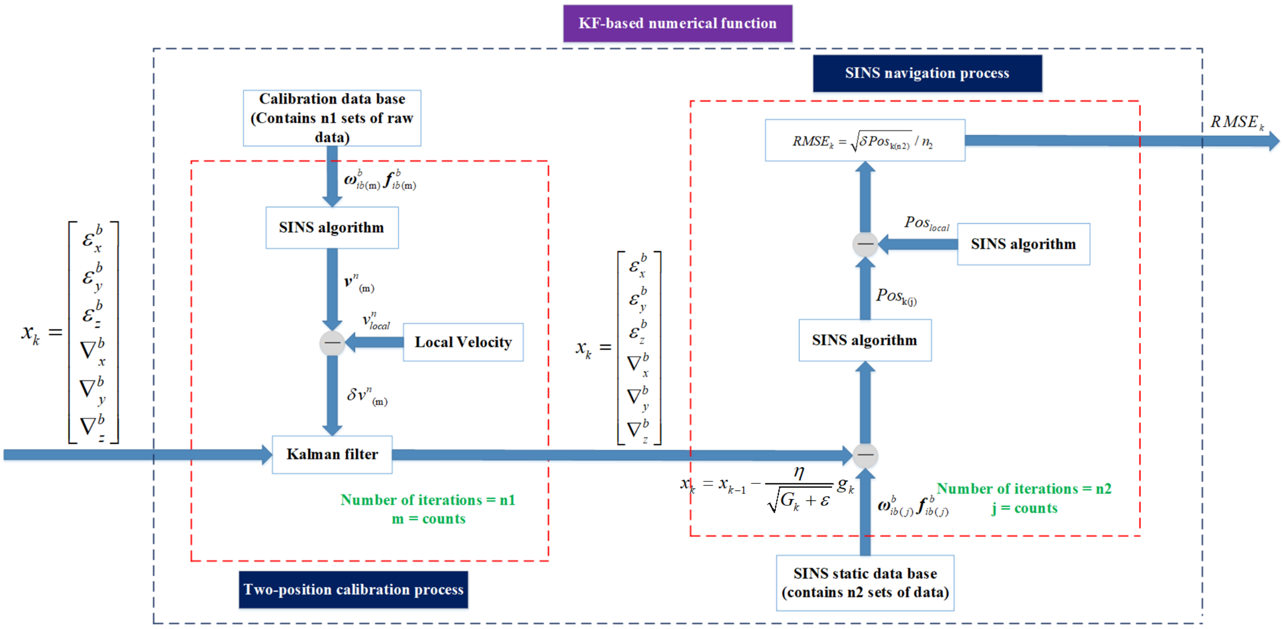
KF-based numerical function
SINS navigation process
Calibration data base (Contains n1 sets of raw data)
$\omega^b_{ib(m)}, f^b_{ib(m)}$
SINS algorithm
$v^n_{(m)}$
$v^n_{local}$
Local Velocity
$\delta v^n_{(m)}$
Kalman filter
Number of iterations = n1
m = counts
Two-position calibration process
$x_k = [\varepsilon^b_x\ \varepsilon^b_y\ \varepsilon^b_z\ \nabla^b_x\ \nabla^b_y\ \nabla^b_z]$
$RMSE_k = \sqrt{\delta Pos_{k(n2)}} / n_2$
$RMSE_k$
$Pos_{local}$
SINS algorithm
$Pos_{k(j)}$
SINS algorithm
$x_k = x_{k-1} - \frac{\eta}{\sqrt{G_k + \varepsilon}} g_k$
$\omega^b_{ib(j)}, f^b_{ib(j)}$
Number of iterations = n2
j = counts
SINS static data base (contains n2 sets of data)

**Figure 2.** KF-based numerical function diagram.

## 4. Analysis of Experimental Tests

For the purpose of verifying the feasibility and evaluating the calibration accuracy of the KF-based AdaGrad algorithm. Both turntable rotation tests and field tests are performed with the KF-based AdaGrad algorithm. The rotation tests were carried out on the three-axis turntable. The sampling rate of the SINS is 4000 Hz, and the ring laser gyros are with an accuracy of 0.01°/h ($1\sigma$) and accelerometers are with 50 $g$ ($1\sigma$).

The gyro signal (lasting 8 h) analysis utilizing Allan variance is shown in Figure 3.

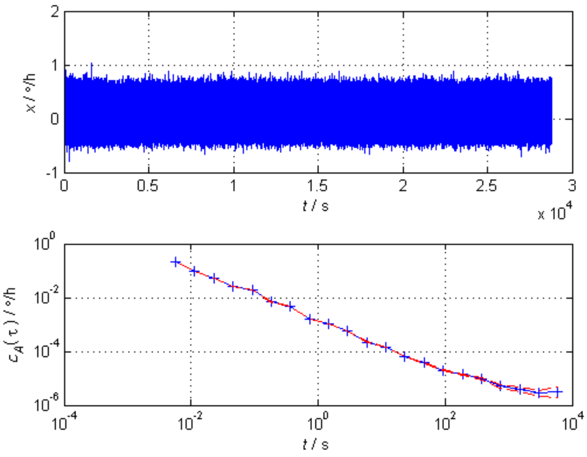

**Figure 3.** Gyroscope signal analysis.

As is shown in Figure 3, the three elements of Allan variance are quantization noise, angle random walk and bias instability. While KF is based on the hypothesis of GWN, the gyros we use usually have irregular noise.

For the purpose of analyzing the influence of the IMU biases on navigation errors. We list two tables to represent the error propagation including long-term navigation and short-term navigation. The long-term navigation error propagation is shown in Table 1 and the short-term navigation error is shown in Table 2.

**Table 1.** Long-term navigation error propagation.

| - | $\boldsymbol{\nabla_E}$ | $\boldsymbol{\nabla_N}$ | $\boldsymbol{\varepsilon_E}$ | $\boldsymbol{\varepsilon_N}$ | $\boldsymbol{\varepsilon_U}$ |
|---|---|---|---|---|---|
| $\delta L$ | $\frac{\nabla_E}{g}c_s s_f$ | $\frac{\nabla_N}{g}(1-c_s c_f)$ | $-\varepsilon_E(\frac{s_e}{\omega_{ie}}-\frac{s_s c_f}{\omega_s})$ | $-\varepsilon_N[\frac{s_L}{\omega_{ie}}(1-c_e)-\frac{s_s s_f}{\omega_s}]$ | $\frac{\varepsilon_U}{\omega_{ie}}c_L(1-c_e)$ |
| $\delta\lambda$ | $\frac{\nabla_E}{g}e_L(1-c_s c_f)$ | $-\frac{\nabla_N}{g}e_L c_s s_f$ | $-\varepsilon_E e_L[\frac{s_L}{\omega_{ie}}(1-c_e)-\frac{s_s s_f}{\omega_s}]$ | $\varepsilon_N(c_L t+\frac{s_L t_L}{\omega_{ie}}s_e-\frac{e_L}{\omega_s}s_s c_f)$ | $\varepsilon_U s_L(t-\frac{1}{\omega_{ie}}s_e)$ |

**Table 2.** Short-term navigation error propagation.

| - | $\boldsymbol{\nabla_E}$ | $\boldsymbol{\nabla_N}$ | $\boldsymbol{\varepsilon_E}$ | $\boldsymbol{\varepsilon_N}$ | $\boldsymbol{\varepsilon_U}$ |
|---|---|---|---|---|---|
| $\delta L$ | 0 | $\nabla_N\frac{t^2}{2R}$ | $-\varepsilon_E\frac{gt^3}{6R}$ | 0 | 0 |
| $\delta\lambda$ | $\nabla_E\frac{t^2}{2Rc_L}$ | 0 | 0 | $\varepsilon_N\frac{gt^3}{6Rc_L}$ | 0 |

As are shown in Tables 1 and 2, the IMU biases would cause navigation error both in long-term navigation and short-term navigation. In order to verify the calibration effect, we need to carry out long-term navigation and short-term navigation experiments.

Through the comparison between Tables 1 and 2. We can find out that in the short-term navigation, the gyro bias of the upward gyro does not affect the navigation accuracy in static condition. The position error of the long-term navigation process is affected by many error sources. Therefore, it is difficult to decompose the influence of IMU errors using conventional decoupling method .

Three car-mounted field tests were carried out in Xi'an, Shaanxi province and Chongqing to verify the effectiveness of the calibration methods. The car-mounted field tests in Xi'an are short-term experiments and the test in Chongqing is a long-term experiment. We utilized a DPGS with an accuracy of 3 cm as the reference position information. Compared with line search and two-position methods, the experimental results can prove the feasibility of the calibration method [32–35].

### *4.1. Static Tests*

According to Section 3, this paper introduces two improved methods for gradient descent algorithm. In order to obtain the minimum value of the KF-based numerical function $\boldsymbol{F}(\boldsymbol{x}_k)$ in a stable way, an algorithm performance analysis among these algorithms is needed.

In this subsection, static tests were utilized to obtain $\boldsymbol{F}(\boldsymbol{x}_k)$. As is shown in Figure 4, SINS is stiffly fixed on a three-axis turntable. After confirming the fixation, calibration is started. The recorded data were saved as a calibration database and SINS static database. The flowchart of AdaGrad and the diagram of KF-based numerical function are also displayed in this paper. Static tests and calibration processes were carried out in a three-axis turntable.

The raw data including three gyros and three accelerometers from the experiment was collected by data collection computer. The collected data was utilized as the calibration database. Afterward, the static tests were carried out for 1 h, the IMU raw data was collected as SINS static database to build the numerical function $\boldsymbol{F}(\boldsymbol{x}_k)$. We utilized the numerical function to compare line search and AdaGrad. Set $\boldsymbol{x}_k$ equals zero vector as the initial value and $RMSE_k$ as the function result. The number of iterations was set as 5000 because too many iterations are too much burden for a normal experiment computer. Results are shown in Table 3.

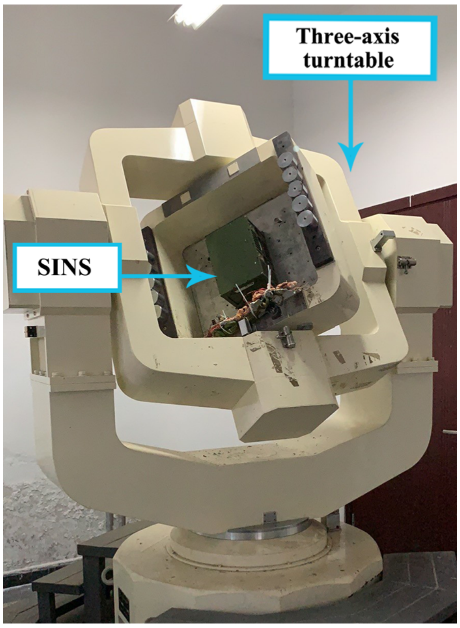

**Figure 4.** Calibration facilities.

**Table 3.** Position RMSE results.

| Iterations | Gradient Descent | Line Search | AdaGrad |
|---|---|---|---|
| 0 | 214.2765 m | 214.2765 m | 214.2765 m |
| 10 | 150.2434 m | 145.7865 m | 134.9809 m |
| 100 | 135.7649 m | 108.9873 m | 90.091 m |
| 200 | 133.2765 m | 87.9087 m | 78.9007 m |
| 1000 | 132.2653 m | 76.9876 m | 22.1107 m |
| 2000 | 132.2653 m | 72.8974 m | 22.1098 m |
| 5000 | 132.2652 m | 70.7477 m | 22.1097 m |

The IMU biases estimation curves utilizing AdaGrad are shown in Figures 5 and 6.

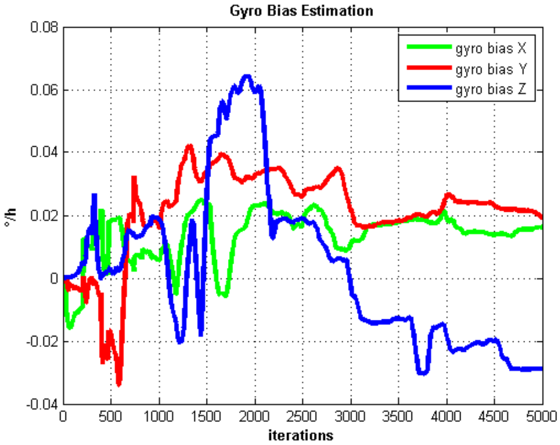

**Figure 5.** Gyro bias estimation.

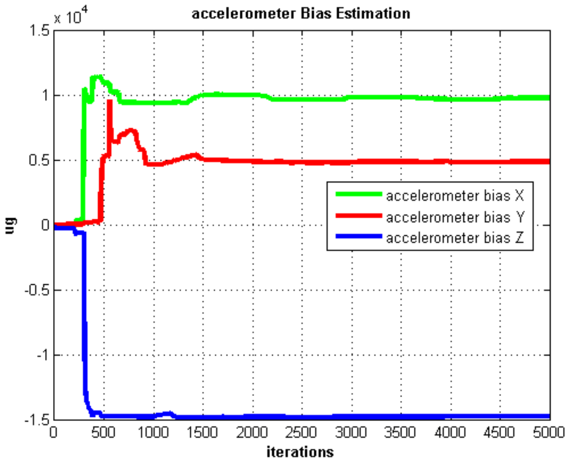

**Figure 6.** Accelerometer bias estimation.

The self-rotation angular rate we set in the static test is 20 °/s. In Figure 5, we can see the gyro biases converge slowly. Gyro biases converge after about 4000 iterations. In contrast, the accelerometer biases converge very fast after about 1000 iterations which are shown in Figure 6.

The static test (lasts 1 h) experimental results are shown in Figures 7–10.

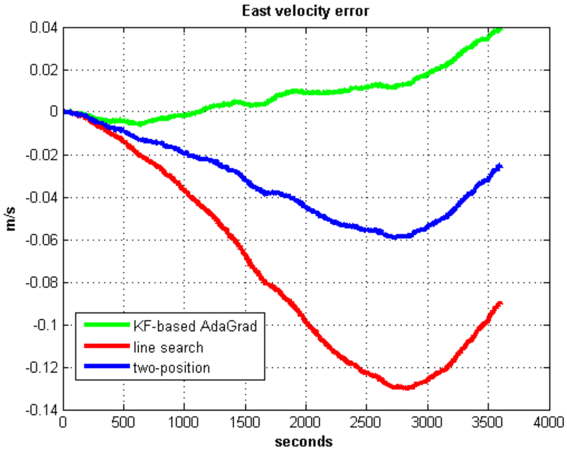

**Figure 7.** East velocity error.

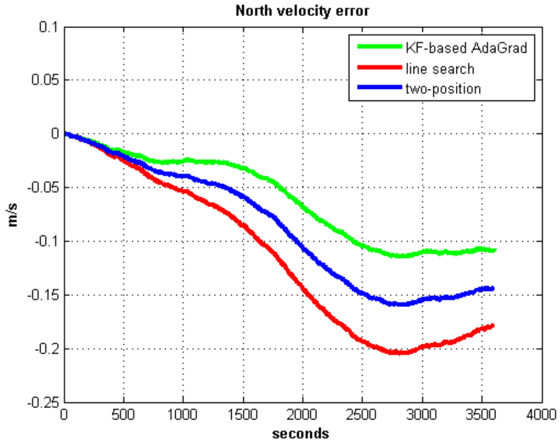

**Figure 8.** North velocity error.

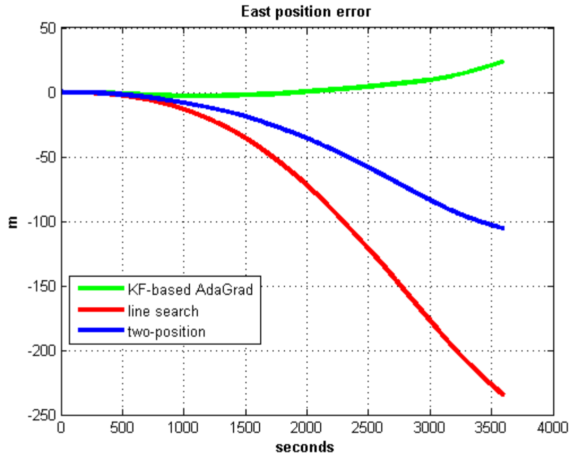

**Figure 9.** East position error.

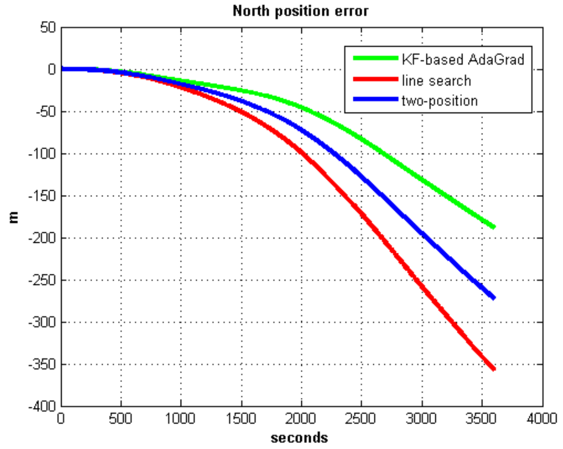

**Figure 10.** North position error.

We do not compare with the gradient descent method because in Table 3, the gradient descent method clearly entered into a locally optimal solution. Experimental results show that the convergence speed of line search is slower than AdaGrad. Due to the limit of iterations, the accuracy of the line search is lower than the AdaGrad method. The velocity and position errors show that the two-position calibration method is more accurate than the line search. AdaGrad has faster convergence rate and higher accuracy in static tests. In order to fully verify the calibration results in dynamic environment, 8 h static tests were carried out. The experimental results were shown in Figures 11–14.

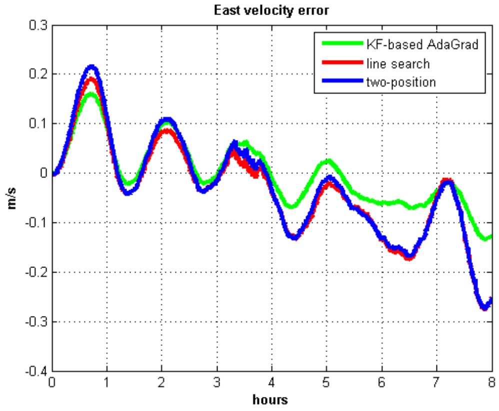

**Figure 11.** East velocity error.

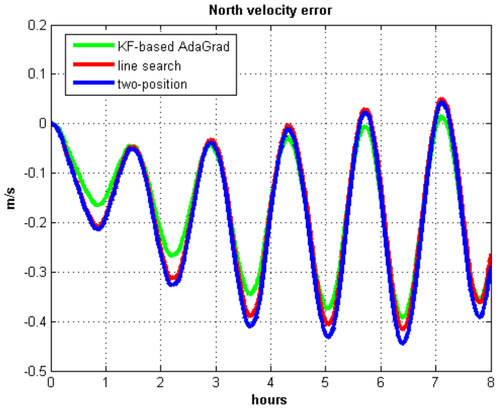

**Figure 12.** North velocity error.

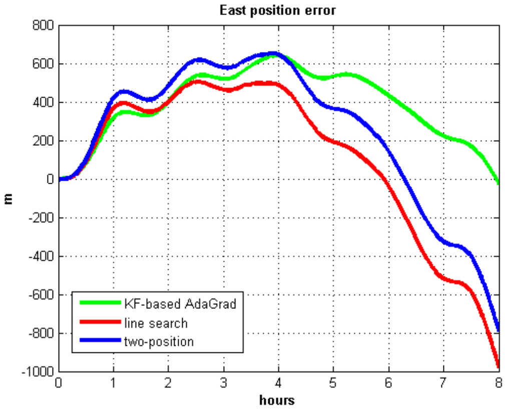

**Figure 13.** East position error.

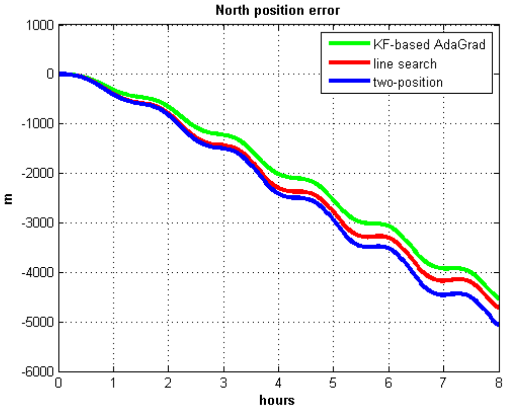

**Figure 14.** North position error.

The velocity and position errors show that KF-based AdaGrad calibration method is more accurate than the line search and the two-position calibration method. The 8 h test shows the peak value of each north and east position errors. AdaGrad has higher accuracy in static test.

### 4.2. *Field Tests*

The static test only considers the angular motion. In order to fully verify the proposed method in this paper, we need to consider the case with linear motion. The field tests were conducted in this subsection. As is shown in Figure 15, the SINS and the DGPS are equipped on the experimental car. The scale factors and installation angles were accurately calibrated before the field tests were carried out.

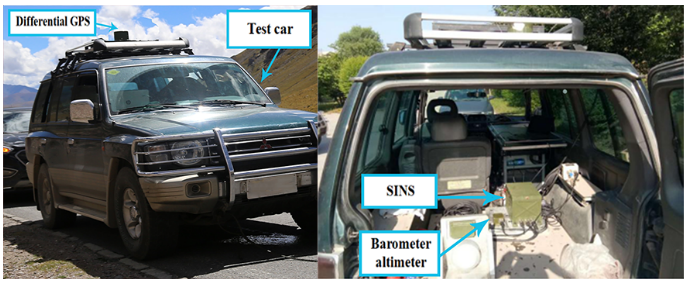

**Figure 15.** Experimental car and equipment.

Three field tests were carried out and the corresponding trajectories are shown in Figure 16. Table 4 shows the field test arrangements. During the maneuvering process of the vehicle, we utilize the SINS/GPS integrated algorithm to provide precise velocity and position information as the velocity and position reference. In order to compare the two-position algorithm, line search and AdaGrad, we utilize the same trajectories to compare the three algorithms.

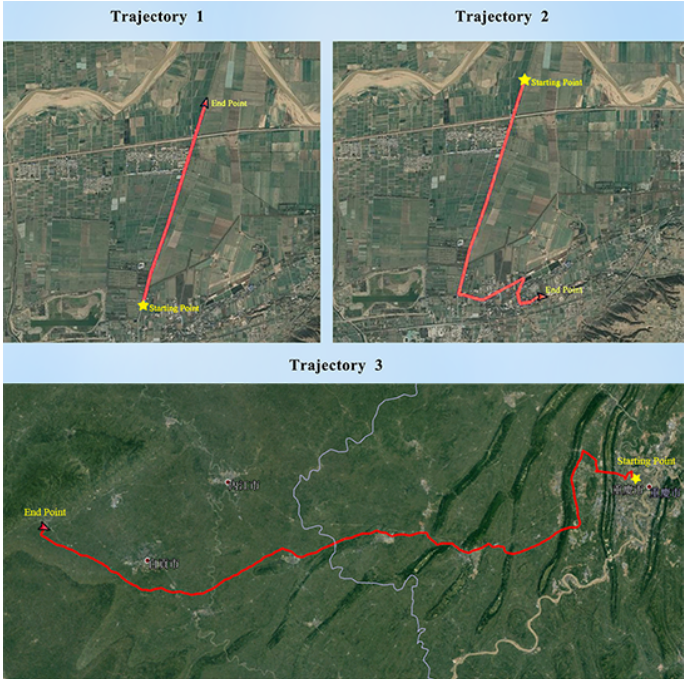

**Figure 16.** Field test trajectories.

**Table 4.** The arrangements of the field tests.

| Field Test Number | Static Stage Time | Maneuvering Stage Time | Total Distance | Maximum Speed |
|---|---|---|---|---|
| Trajectory 1 | 300 s | 651 s | 3560.52 m | 14.72 m/s |
| Trajectory 2 | 300 s | 768 s | 5870.08 m | 16.65 m/s |
| Trajectory 3 | 900 s | 8 h | 267.98 km | 30.65 m/s |

Field test 1 is a straight-line road test and field test 2 contained several turns and one sharp turn. The test vehicle had no obstruction on the test roads thus the GPS signal is consistent and effective thus the SINS/DGPS integrated algorithm can provide stable position information and velocity information.

The results of field test 1 are shown in Figures 17–20.

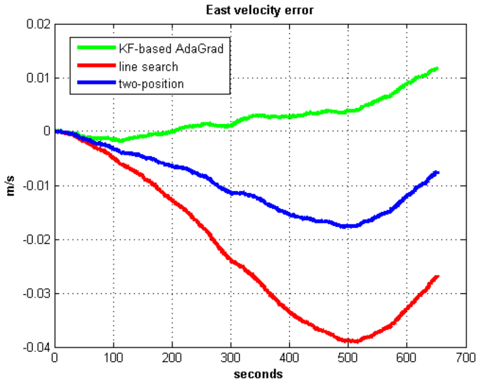

**Figure 17.** East velocity error in test 1.

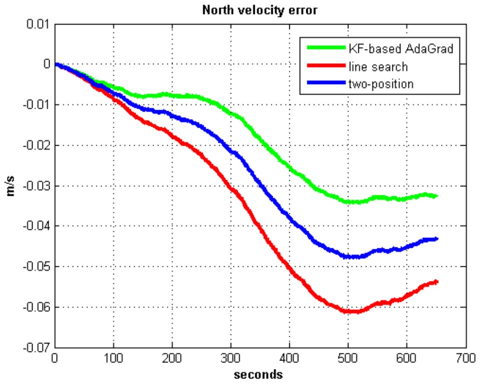

**Figure 18.** North velocity error in test 1.

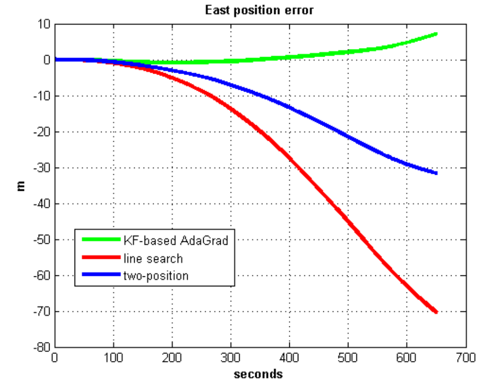

**Figure 19.** East position error in test 1.

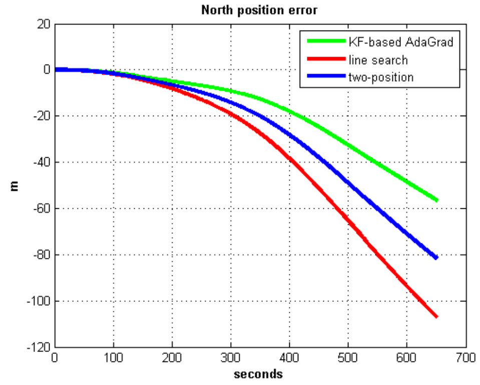

**Figure 20.** North position error in test 1.

We list the RMS and peak value of velocity and position error of test 1 in Tables 5 and 6.

**Table 5.** RMS of velocity and position errors.

| - | Line Search | Two-Position | KF-Based AdaGrad |
|---|---|---|---|
| $\delta v_E$ | 0.005982 m/s | 0.002265 m/s | 0.001276 m/s |
| $\delta v_N$ | 0.01055 m/s | 0.008769 m/s | 0.005989 m/s |
| $\delta p_E$ | 26.6124 m | 11.4123 m | 5.1241 m |
| $\delta p_N$ | 32.9655 m | 27.5421 m | 21.4231 m |

**Table 6.** Maximum errors of velocity and position.

| - | Line Search | Two-Position | KF-Based AdaGrad |
|---|---|---|---|
| $\delta v_E$ | −0.0389 m/s | −0.0178 m/s | 0.0132 m/s |
| $\delta v_N$ | −0.0613 m/s | −0.0489 m/s | −0.0311 m/s |
| $\delta p_E$ | −71.6436 m | −31.2141 m | 8.7231 m |
| $\delta p_N$ | −108.0921 m | −81.9215 m | −58.9249 m |

In field test 1, Figures 17 and 18 shows that the east and north velocity errors of which the utilized KF-based AdaGrad are the smallest among the three methods. The two-position method is more accurate than the line search method. The position error is similar to the velocity error. Figures 19 and 20 show that the east and north position error which utilized the two-position method smaller than the error of which utilized line search. The method utilized KF-based AdaGrad has the smallest position error.

The results of field test 2 are shown in Figures 21–24.

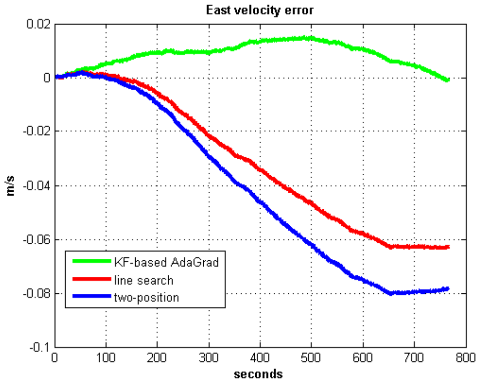

**Figure 21.** East velocity error in test 2.

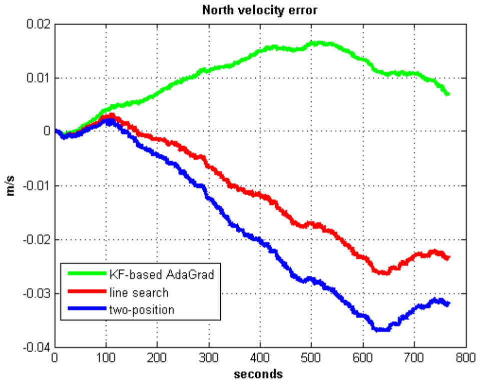

**Figure 22.** North velocity error in test 2.

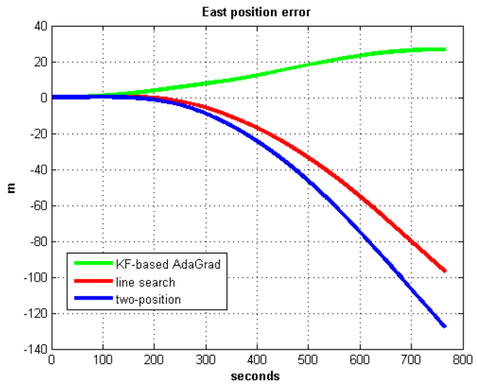

**Figure 23.** East position error in test 2.

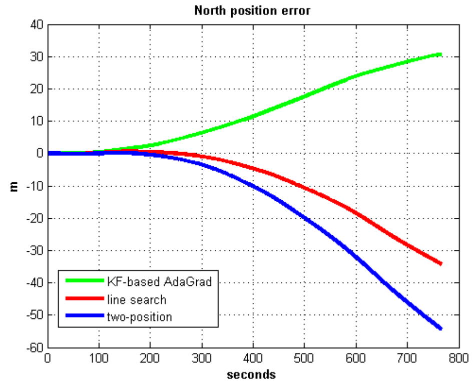

**Figure 24.** North position error in test 2.

In field test 2, Figures 21 and 22 show that the east and north velocity error utilizing KF-based AdaGrad is the smallest among the three methods. The line search method is more accurate than the two-position method. The position errors are shown in Figures 23 and 24, the east and north position error which utilized KF-based AdaGrad are obviously smaller than the other two methods. The position error of which utilized line search is smaller than the error of which utilized two-position method. The results of RMS and maximum values of velocity and position error are shown in Tables 7 and 8.

**Table 7.** RMS of velocity and position errors.

| - | Line Search | Two-Position | KF-Based AdaGrad |
|---|---|---|---|
| $\delta v_E$ | 0.03982 m/s | 0.04123 m/s | 0.01076 m/s |
| $\delta v_N$ | 0.01255 m/s | 0.01476 m/s | 0.010989 m/s |
| $\delta p_E$ | 35.2412 m | 38.5124 m | 11.5613 m |
| $\delta p_N$ | 16.6278 m | 21.0138 m | 13.4163 m |

**Table 8.** Maximum errors of velocity and position.

| - | Line Search | Two-Position | KF-Based AdaGrad |
|---|---|---|---|
| $\delta v_E$ | −0.0613 m/s | −0.0809 m/s | 0.0169 m/s |
| $\delta v_N$ | −0.0378 m/s | −0.0276 m/s | 0.0165 m/s |
| $\delta p_E$ | −127.2532 m | −97.5235 m | 23.9873 m |
| $\delta p_N$ | −33.8349 m | −53.6821 m | 31.3452 m |

For the purpose of fully verifying the effectiveness of the KF-based AdaGrad calibration method in the long-term navigation process, an 8 h field test was carried out in

Chongqing to excite all three parts with different periods and amplitude. Due to the unstable GPS signal in field test 3, we only compare position errors because of the low accuracy of GPS velocity. The 8 h position error curve can fully verify the estimated accuracy of IMU biases.

The results of field test 3 are shown in Figures 25 and 26.

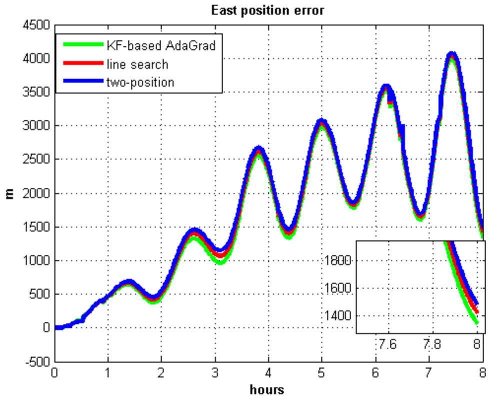

**Figure 25.** East position error in test 3.

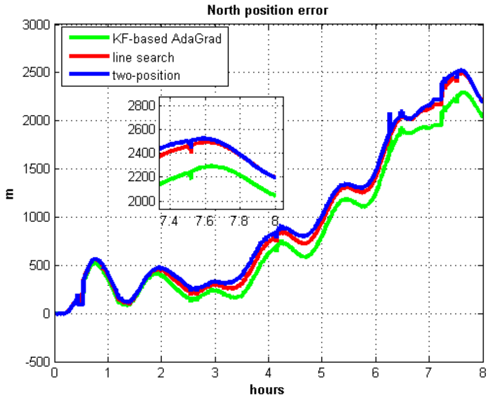

**Figure 26.** North position error in test 3.

In field test 3, the east and north position errors that utilized KF-based AdaGrad are smaller than the other two methods. Under long maneuvering condition, the line search and two-position calibration method have similar estimation accuracy.

The peak value of Figures 25 and 26 is easy to figure out, we need to analyze the RMS of the position error. Table 9 shows the RMS of position errors in field test 3.

**Table 9.** RMS of position errors.

| - | Line Search | Two-Position | KF-Based AdaGrad |
|---|---|---|---|
| $\delta p_E$ | 1789.5758 m | 1745.7639 m | 1714.8437 m |
| $\delta p_N$ | 987.0832 m | 956.8051 m | 914.5033 m |

The results differences between field test 1 and field test 2 are the line search and two-position method because test 2 is more dynamic than test 1. Navigation errors due to IMU biases can be excited out more easily so the convergence rate of line search is faster in test 2. The AdaGrad has the highest accuracy among the three methods. In conclusion, under the severe maneuvering condition line search has higher accuracy than two-position method while under low maneuvering conditions the traditional method two-position is more accurate than line search. Both static tests and car-mounted tests show the proposed method can achieve a more accurate estimated velocity and position than the other two methods, which denotes that the KF-based AdaGrad can estimate the IMU biases more accurately. The convergence rate and accuracy of line search are greatly affected by the external dynamic condition. Thus, the line search is not fittable in IMU biases calibration.

## 5. Conclusions

The characteristics of IMU are usually difficult to determine. The IMU biases severely affecting navigation accuracy. In order to solve this problem, in this paper, we propose a KF-based AdaGrad calibration algorithm. We derive a KF-based AdaGrad numerical function and utilized AdaGrad method to solve the problem of setting step size. The recorded data were saved as the calibration database and the SINS static database. The flowchart of AdaGrad and the diagram of KF-based numerical function are also displayed in this paper. Static tests and calibration process were carried out in a three-axis turntable and three field tests were carried out in Xi'an. Experimental results show that the proposed calibration method can effectively improve the accuracy of estimation of IMU biases in both static and dynamic conditions.

**Author Contributions:** Conceptualization, Z.W.; methodology, Z.W.; software, Z.W.; validation, Q.C.; formal analysis, Z.W.; investigation, Z.W.; resources, Q.C.; data curation, Q.C.; writing—original draft preparation, Z.W.; writing—review and editing, Q.C.; visualization,G.Y.; supervision, G.Y.; project administration, G.Y.; funding acquisition, Q.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (61803015).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Huang, Y.; Zhang, Y. A new process uncertainty robust student's t based Kalman filter for SINS/GPS integration. *IEEE Access* **2017**, *5*, 14391–14404. [CrossRef]
2. Qi, N. Research on initial alignment of SINS for marching vehicle. In Proceedings of the 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, Shenyang, China, 21–23 September 2013; pp. 465–468.
3. Tedaldi, D.; Pretto, A.; Menegatti, E. A robust and easy to implement method for IMU calibration without external equipments. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3042–3049.
4. Fong, W.T.; Ong, S.K.; Nee, A.Y.C. Methods for in-field user calibration of an inertial measurement unit without external equipment. *Meas. Sci. Technol.* **2008**, *19*, 085202. [CrossRef]
5. Wu, Y.; Hu, D.; Wu, M.; Hu, X.; Wu, T. Observability analysis of rotation estimation by fusing inertial and line-based visual information: A revisit. *Automatica* **2006**, *42*, 1809–1812. [CrossRef]
6. Li, J.; Fang, J.; Du, M. Error analysis and gyro-bias calibration of analytic coarse alignment for airborne POS. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 3058–3064.
7. Han, X.; Hu, S.M.; Luo, H. A new method for bias calibration of laser gyros using a single-axis turning table. *Optik* **2013**, *124*, 5588–5590. [CrossRef]
8. Zheng, Z.; Han, S.; Zheng, K. An eight-position self-calibration method for a dual-axis rotational Inertial Navigation System. *Sens. Actuators A Phys.* **2015**, *232*, 39–48. [CrossRef]
9. Wang, H.; Deng, Z.; Feng, B.; Ma, H.; Xia, Y. An adaptive Kalman filter estimating process noise covariance. *Neurocomputing* **2017**, *223*, 12–17. [CrossRef]
10. Gao, Z.; Mu, D.; Gao, S.; Zhong, Y.; Gu, C. Adaptive unscented Kalman filter based on maximum posterior and random weighting. *Aerosp. Sci. Technol.* **2017**, *71*, 12–24. [CrossRef]
11. Jurman, D.; Jankovec, M.; Kamnik, R.; Topič, M. Calibration and data fusion solution for the miniature attitude and heading reference system. *Sens. Actuators A Phys.* **2007**, *138*, 411–420. [CrossRef]
12. Munguia, R.; Grau, A. Attitude and Heading System based on EKF total state configuration. In Proceedings of the 2011 IEEE International Symposium on Industrial Electronics, Gdansk, Poland, 27–30 June 2011; pp. 2147–2152.
13. Li, W.; Wang, J. Effective adaptive kalman filter for MEMS-IMU/magnetometers integrated attitude and heading reference systems. *J. Navig.* **2013**, *66*, 99–113. [CrossRef]
14. Hyyti, H.; Visala, A. A DCM Based Attitude Estimation Algorithm for Low-Cost MEMS IMUs. *Int. J. Navig. Obs.* **2015**, *2015*, 503814. [CrossRef]
15. Wang, Q.; Gao, C.; Wei, G.; Long, X. Online calibration technique for LDV in SINS/LDV integrated navigation systems. *Sens. Command. Control. Commun. Intell. Technol. Homel. Secur. Defense Law Enforc. Appl. XVI* **2017**, *10184*, 1018415.

16. Li, K.; Chen, Y.; Wang, L. Online self-calibration research of single-axis rotational inertial navigation system. *Meas. J. Int. Meas. Confed.* **2018**, *129*, 633–641. [CrossRef]
17. Huang, H.; Zhou, J.; Zhang, J.; Yang, Y.; Song, R.; Chen, J.; Zhang, J. Attitude Estimation Fusing Quasi-Newton and Cubature Kalman Filtering for Inertial Navigation System Aided with Magnetic Sensors. *IEEE Access* **2018**, *6*, 28755–28767. [CrossRef]
18. He, H.; Zhu, B.; Zha, F. Particle Swarm Optimization-Based Gyro Drift Estimation Method for Inertial Navigation System. *IEEE Access* **2019**, *7*, 55788–55796. [CrossRef]
19. Jinwu, T.; Xiaosu, X.; Tao, Z.; Liang, Z.; Yao, L. Study on Installation Error Analysis and Calibration of Acoustic Transceiver Array Based on SINS/USBL Integrated System. *IEEE Access* **2018**, *6*, 66923–66939. [CrossRef]
20. Rajamani, M.R.; Rawlings, J.B. Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica* **2009**, *45*, 142–148. [CrossRef]
21. Sheimy, N.E.I.; Hou, H.; Niu, X. Analysis and Modeling of Inertial Sensors Using AV. *IEEE Trans. Instrum. Meas.* **2008**, 56–57.
22. Ford, J.J.; Evans, M.E. On-line estimation of Allan variance parameters. *J. Guid. Control. Dyn.* **2000**, *23*, 980–987. [CrossRef]
23. Yu, H.; Wu, W.; Wu, M.; Feng, G.; Hao, M. Systematic angle random walk estimation of the constant rate biased ring laser Gyro. *Sensors* **2013**, *13*, 2750–2762. [CrossRef]
24. Axelsson, H.; Wardi, Y.; Egerstedt, M.; Verriest, E.I. Gradient descent approach to optimal mode scheduling in hybrid dynamical systems. *J. Optim. Theory Appl.* **2008**, *136*, 167–186. [CrossRef]
25. Fliege, J.; Svaiter, B.F. Steepest descent methods for multicriteria optimization. *Math. Methods Oper. Res.* **2000**, *51*, 479–494. [CrossRef]
26. Wang, M.; Fang, E.X.; Liu, H. Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions. *Math. Program.* **2017**, *161*, 419–449. [CrossRef]
27. Luo, B.; Liu, D.; Huang, T.; Yang, X.; Ma, H. Multi-step heuristic dynamic programming for optimal control of nonlinear discrete-time systems. *Inf. Sci.* **2017**, *411*, 66–837. [CrossRef]
28. Zhang, H.; Niu, Y.; Lu, J.; Yang, Y. System-level calibration for the star sensor installation error in the stellar-inertial navigation system on a swaying base. *IEEE Access* **2018**, *6*, 47288–47294. [CrossRef]
29. Zhang, J.; Li, J.; Huang, Y.; Hu, C.; Feng, K.; Wei, X. Analysis and Compensation of Installation Errors for Rotating Semi-Strapdown Inertial Navigation System. *IEEE Access* **2019**, *7*, 101019–101030. [CrossRef]
30. Chang, L.; Qin, F.; Jiang, S. Strapdown Inertial Navigation System Initial Alignment Based on Modified Process Model. *IEEE Sens. J.* **2019**, *19*, 6381–6391. [CrossRef]
31. Gao, P.; Li, K.; Wang, L.; Gao, J. A Self-Calibration Method for Non-Orthogonal Angles of Gimbals in Tri-Axis Rotational Inertial Navigation System. *IEEE Sens. J.* **2016**, *16*, 8998–9005. [CrossRef]
32. Lu, J.; Liu, X.; Zhang, R. Calibration, alignment, and dynamic tilt maintenance method based on vehicular hybrid measurement unit. *IEEE Sens. J.* **2019**, *19*, 7243–7253. [CrossRef]
33. Wen, Z.; Yang, G.; Cai, Q.; Sun, Y. Modeling and Calibration of the Gyro- Accelerometer Asynchronous Time in Dual-Axis RINS. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–17. [CrossRef]
34. Xu, X.; Gui, J.; Sun, Y.; Yao, Y.; Zhang, T. A Robust In-Motion Alignment Method with Inertial Sensors and Doppler Velocity Log. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
35. Xu, X.; Sun, Y.; Gui, J.; Yao, Y.; Zhang, T. A Fast Robust In-Motion Alignment Method for SINS with DVL Aided. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3816–3827. [CrossRef]